THIS PAGE BLANK (USPTO)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 912 063 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
28.04.1999 Bulletin 1999/17

(51) Int. Cl.^{6}: **H04N 7/50**

(21) Application number: 98119908.6

(22) Date of filing: 21.10.1998

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 24.10.1997 SG 9703861

(71) Applicant:
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.
Kadoma-shi, Osaka 571-8501 (JP)

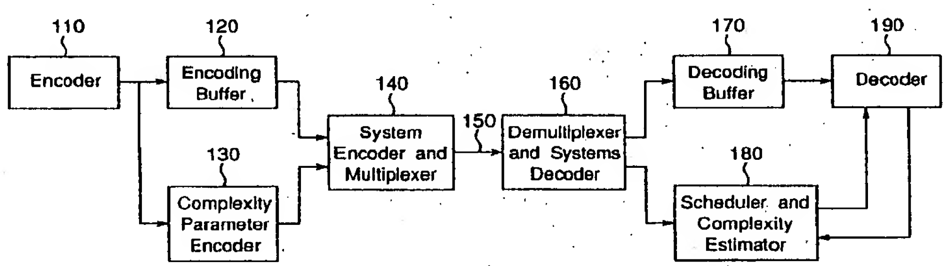
(72) Inventor: **Tan, Thiow Keng**
08-506 Singapore 470601 (SG)

(74) Representative:
Eisenführ, Speiser & Partner
Martinistrasse 24
28195 Bremen (DE)

(54) A method for computational graceful degradation in an audiovisual compression system

(57) The invention disclosed here is a method for an encoder to encode audiovisual information for transmission to the decoder without any prior knowledge of the computational capabilities of the decoder. A descriptor containing parameters that can be used to estimate the complexity of the decoding process is embedded in the system stream. The encoder also encodes the video information in such a manner that the decoder can choose to ignore some of the information and only decode a subset of the encoded information in order to reduce the computational requirements. This method allows more than one decoder to decode the same bitstream giving different resolutions depending on the computational capability of the decoder.

Fig. 1



EP 0 912 063 A2

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates to a method for computational graceful degradation in an audiovisual compression system. This invention is useful in a multimedia encoding and decoding environment where the computational demands for decoding a bitstream is not well defined. It is also useful in cases where channel capacity is limited and some form of quality of service guarantee is required. It is also useful for inter working between two video services of different resolutions.

2. Description of the Related Art

[0002] It is common in the case of software decoding to employ some form of graceful degradation when the system resources is not sufficient to fully decode all of the video bitstream. These degradation ranges from partial decoding of the picture elements to dropping of complete pictures. This is easy to implement in the case of a single video stream.

[0003] In the proposed new ISO/IEC SC29/WG11 standard of MPEG-4, it is possible to send multiple Audiovisual, AV, objects. Therefore, the total complexity requirements no longer depend on one single stream but on multiple streams.

[0004] In compression systems such as MPEG-1, MPEG-2 and MPEG-4, a high degree of temporal redundancy is removed by employing motion compensation. It is intuitive to see that successive pictures in a video sequence will contain very similar information. Only regions of the picture that are moving will change from picture to picture. Furthermore, these regions usually move as a unit with uniform motion. Motion compensation is a technique where the encoder and the decoder keep the reconstructed picture as a reference for the prediction of the current picture being encoded or decoded. The encoder mimics the decoder by implementing a local decoder loop. Thus, keeping the reconstructed picture synchronized between the encoder and decoder.

[0005] The encoder performs a search for a block in the reconstructed picture that gives the closest match to the current block that is being encoded. It then computes the prediction difference between the motion compensated block and the current block being encoded. Since the motion compensated block is available in the encoder and the decoder, the encoder only needs to send the location of this block and the prediction difference to the decoder. The location of the block is commonly referred to as the motion vector. The prediction difference is commonly referred to as the motion compensated prediction error. These information requires less bits to send that the current block itself.

[0006] In intra-picture coding, spatial redundancy may be removed in a similar way. The transform coefficients of the block can be predicted from the transform prediction of its neighboring blocks that have already being decoded.

[0007] There are two major problems to be solved in this invention. The first is how to indicate the decoding complexity requirements of the current AV object. In the case where there are multiple AV objects, the systems decoder must decide how much resource should be given to a particular object and which object should have priority over another. In other words, how to model the complexity requirements of the system. A point to be noted here is that the complexity requirements of the decoder is dependent on the implementation of the decoder. An operation that is complex for one implementation may be simple for another implementation. Therefore, some form of implementation independent complexity measure is required.

[0008] The second problem is how to reduce complexity requirements in the decoder. This deals with the method of reducing the complexity requirements of the decoding process while retaining as much of the information as possible. One biggest problem in graceful degradation is the problem of drift caused by errors in the motion compensation. When graceful degradation is employed the reconstructed picture is incomplete or noisy. These errors are propagated from picture to picture resulting in larger and larger errors. This noise propagation is referred to as drift.

SUMMARY OF THE INVENTION

[0009] In order to solve the problems the following steps are taken in the present invention.

[0010] The AV object encoder encodes the AV object in a manner that would allow different amounts of graceful degradation to be employed in the AV object decoder. Parameters relating to the computational complexity requirements of the AV objects are transmitted in the systems encoder. Implementation independent complexity measure is achieved by sending parameters that gives an indication of the operations that are required.

[0011] At the systems decoder, estimates of the complexity required are made based on these parameters as well as the implementation methods being employed. The resource scheduler then allocates the appropriate amount of resources to the decoding of the different AV objects. In the AV object decoder, computational graceful degradation is employed when the resources are not sufficient to decode the AV object completely.

[0012] In accordance with a first aspect of the present invention, a method of encoding a plurality of audiovisual objects into a compressed coded representation suitable for computational graceful degradation at the decoder comprises:

encoding said audiovisual objects, incorporation methods allowing computational graceful degradation to be employed in the decoder, into their coded representations;

estimating the implementation independent computational complexity measures in terms of a plurality of block decoding parameters;

partitioning said coded representations of the audiovisual objects into a plurality of access units and adding header information to form packets;

inserting a descriptor containing said block decoding parameters into the header of the packet; and

multiplexing these packets to form a single multiplexed bitstream.

[0013] In accordance with a second aspect of the present invention, a method of decoding a multiplexed bitstream, with computational graceful degradation, to obtained a plurality of audiovisual objects, comprises;

de-multiplexing the single multiplexed bitstream into a plurality of packets comprising of packet headers and access units;

extracting the descriptor containing a plurality of block decoding parameters from the packet headers;

reassembling the access units into their original coded representations of the audiovisual objects;

estimating the decoder specific computational complexity measures based on said block decoding parameters and the current decoder implementation; and

decoding said coded representations of the audiovisual objects, using computational graceful degradation, where necessary, to satisfy the estimated decoder specific computational complexity requirements.

[0014] Preferably, the incorporation methods allowing computational graceful degradation to be employed in the decoder, comprise;

partitioning the input pictures to be encoded into a plurality of sub-regions numbered in increasing order, beginning with the full picture as the first sub-region, where each sub-region comprising only of a subset of the pixels within the sub-region preceding it;

entropy coding the position and dimension of the sub-regions into a compressed coded representation within the bitstream;

further partitioning the sub-regions into a plurality of blocks for encoding into a compressed coded representation within the bitstream;

performing motion estimation and motion compensation for said blocks using only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks;

entropy coding the motion vectors into a compressed coded representation within the bitstream;

transforming the motion compensated prediction difference into an orthogonal domain;

quantizing the transformed coefficients using a quantization method; and

entropy coding the quantized transformed coefficients into a compressed coded representation within the bitstream.

[0015] Preferably, the method for decoding the coded representations of the audiovisual objects in accordance with the second aspect, using computational graceful degradation, where necessary to satisfy the estimated decoder specific computational complexity requirements, further comprises:

entropy decoding the position and dimension of the sub-regions from the compressed coded representation within the bitstream;

selecting only the blocks that are within the sub-region of interest for decoding;

entropy decoding the compressed coded representation to give quantized transformed coefficients;

inverse quantizing said quantized transformed coefficients to give the transformed coefficients;

inverse transforming said transform coefficients to give the spatial domain motion compensated prediction difference;

entropy decoding the motion vectors from the compressed coded representation within the bitstream;

performing motion compensation for said blocks using only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks; and,

reconstructing the picture and storing said picture in the frame memory for prediction of the next picture.

[0016] Preferably, the method in accordance with the first aspect of the invention, whereby incorporation methods allowing computational graceful degradation to be employed in the decoder, further comprises:

partitioning the input pictures to be encoded into a plurality of sub-regions numbered in increasing order, beginning with the full picture as the first sub-region, where each sub-region comprising only of a subset of the pixels within the sub-region preceding it;

entropy coding the position and dimension of the sub-regions into a compressed coded representation within the bitstream;

further partitioning the sub-regions into a plurality of blocks for encoding into a compressed coded representation within the bitstream;

transforming said blocks into an orthogonal

domain;

quantizing the transformed coefficients using a quantization method;

performing quantized transform coefficient prediction for said blocks using only the corresponding quantized transform coefficients from the blocks above and to the left that belong to sub-regions having the same or higher numeric order as said blocks; and,

entropy coding the predicted difference of the quantized transformed coefficients into a compressed coded representation within the bitstream.

[0017] Preferably, the method in accordance with the first aspect of the invention, comprises:

entropy decoding the position and dimension of the sub-regions from the compressed coded representation within the bitstream;

selecting only the blocks that are within the sub-region of interest for decoding;

entropy decoding the compressed coded representation to give quantized transformed coefficients;

performing quantized transform coefficient prediction for said blocks using only the corresponding quantized transform coefficients from the blocks above and to the left that belong to sub-regions having the same or higher numeric order as said blocks;

inverse quantizing said quantized transformed coefficients to give the transformed coefficients;

inverse transforming said transform coefficients to give the spatial domain pixel values; and,

reconstructing the picture and storing said picture in the frame memory for prediction of the next picture.

[0018] Typically, the plurality of block decoding parameters comprises numeric numbers indicating the number of:

block entropy decoding operations;

block motion compensation operation;

block inverse quantization operations;

block transform operations;

block addition operations; and,

block memory access operations.

[0019] Preferably, the descriptor comprises:

a descriptor identification number signaling the descriptor type;

a descriptor length field to indicate the size of the descriptor; and,

a plurality of block decoding parameters.

[0020] Typically, in the method of partitioning the input pictures to be encoded into a plurality of sub-regions, the sub-regions are rectangular.

[0021] Preferably, in the method of performing motion estimation and motion compensation for said blocks, using only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks, implies that only prediction blocks that lie completely within said sub-regions are selected.

[0022] Typically, when only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks are used, prediction blocks may lie partially outside said sub-regions but with the additional condition that the pixels lying outside said sub-region are replaced by the nearest pixels from within the sub-regions.

[0023] Preferably, in the method of partitioning the pictures into a plurality of sub-regions, the position and dimension of each of said sub-regions may vary from picture to picture and said position and said dimension are coded by means of a pan scan vector, giving the horizontal and vertical displacement, a width and a height.

[0024] Typically, in the method of partitioning the pictures into a plurality of sub-regions, the position and dimension of the sub regions are the same from picture to picture and said position and said dimension are coded once at the beginning of the sequence by means of a horizontal and vertical displacement, a width and a height.

[0025] Preferably, in the method of encoding and decoding, the transform is the Discrete Cosine Transform.

[0026] Typically, in the method of encoding and decoding, the number of sub-regions is two.

[0027] Preferably, in the method where there is a plurality of sub-region numbered in increasing order and the motion vector can point into a sub-region of lower order but not out of a lower order to a higher ordered number.

BRIEF DESCRIPTION OF THE DRAWINGS

[0028]

Figure 1 is an overall block diagram of the present invention;

Figure 2 shows a block diagram of encoder and decoder of the present invention;

Figure 3 illustrates the embodiment of the sub-region and the motion vector restriction of the present invention;

Figure 4 illustrates the embodiment for the pan-scan vectors and the sub-region dimensions in the present invention;

Figure 5 illustrates the second embodiment for the padding method of the motion compensated prediction at the sub-region boundary; and,

Figure 6 illustrates the block diagram for the Com-

plexity Estimator.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0029] Figure 1 shows an overall system block diagram of the present invention. Encoder unit 110 encodes the video sequence to allow computational graceful degradation techniques. The output of encoder 110 is a coded representation of the video sequence that is applied to an encoding buffer 120. At the same time the video sequence and the coded representation are also applied to a complexity parameter encoder 130 where the parameters associated with the operation that are required for decoding is computed and encoded. These information together with the output of the encoding buffer 120 are passed to a System Encoder and Multiplexer unit 140 where a system-multiplexed stream is formed. The system-multiplexed stream is transmitted through a transmission media 150.

[0030] A Demultiplexer and System Decoder unit 160 receives the system-multiplexed stream, where the bitstream is demultiplexed into its respective elementary streams. The video elementary stream is passed to a Decoding Buffer 170, and complexity parameters are passed to a Scheduler and Complexity Estimator unit 180. From the Decoding Buffer 170, the video elementary stream is passed to a Decoder unit 190. The decoder 190 waits for the commands coming from the Scheduler unit 180 before decoding.

[0031] The Complexity Estimator 180 gives the amount of decoder computational graceful degradation that is to be employed. Computational graceful degradation is achieved in the decoder by decoding only a sub-region of the complete picture that is deemed to contain the more important information. The encoder will have to prevent the encoder and decoder from drifting apart under these conditions. After decoding, the decoder unit 190 also feedback information to the Scheduler and Complexity Estimator 180 so that the information may be used to estimate the complexity of the next picture.

[0032] The following is the embodiment of the various units illustrated in the above invention shown in Figure 1.

[0033] Figure 2 is a block diagram of the encoder and decoder according to the present embodiment. The input picture to the encoder 110 is segmented into blocks for processing. Temporal redundancy is removed from the picture by subtracting the motion compensated picture of the previous picture from the current picture. The prediction difference is then transformed into the DCT domain in a DCT unit 111. The resulting DCT coefficients are then quantized in a Quantization unit 112. The quantized coefficients are then entropy coded in a Variable Length Coding (VLC) unit 113 to form the compressed output bitstream. The encoder 110 also has a local decoder loop comprising of an Inverse Quantization unit 114, an Inverse DCT unit 115, a Frame Storage 116, and a Motion Compensation unit 117. The local decoder loops mimics the decoder operations by inverse quantizing the coefficients and transforming it back into the spatial domain in the Inverse Quantization unit 114 and Inverse DCT unit 115. The output is then added to the output of the Motion Compensated unit 117 to form the reconstructed picture. This picture is stored in the Frame Storage 116 for motion compensation of the next picture.

[0034] In this embodiment the encoder units of Motion Estimation unit 118 and Motion Compensation unit 117 are changed so that computational graceful degradation may be performed in conjunction with the motion compensation without causing drift.

[0035] Figure 3 illustrates the present invention, according to which the picture is divided into two parts 220 and 210. The first part 220 is a sub-region that must be decoded in the decoder regardless of whether computational graceful degradation is employed or not. The second part 210 is the region outside of the sub-region, which may be discarded by the decoder when computational graceful degradation is employed.

[0036] Figure 3 also show two blocks that are used for motion compensation. When motion compensation is performed on a block 250 that resides in the sub-region 220, the motion compensated prediction block must also come from within the sub-region 220 of the reference picture. In other words the motion vector 260 pointing out of the region is not allowed. This is referred to restricted motion vector. On the other hand, when a block 230 resides outside the sub-region 220, the motion compensated prediction block can come from anywhere in the reference picture. This is the same as where there is no sub-region.

[0037] Figure 4 shows a method how to indicate the sub-region 220 within each picture. In order to specify the rectangular sub-region 220 for each picture the following parameters must be specified for each picture and be encoded in the picture header of the compress bitstream. In Figure 4, a picture 310 and the sub-region 220 is illustrated. The horizontal offset 330 of the left edge of sub-region 220 from the left edge of the picture, and the vertical offset 340 of the top edge of the sub-region 220 from the top edge of the picture are shown. These two parameters, referred to as the pan scan vectors, are used to indicate the location of the sub-region. The width 350 and the height 360 of the sub-region 220 are the second set of parameters that are required to specify the dimensions of the sub-region 220.

[0038] In a second embodiment of this invention, the motion vector for a block in the sub-region need not be restricted. It is allowed to point out of the sub-region of the reference picture. However padding is required. This is illustrated in Figure 5 in which the picture 310 and the sub-region 220 are shown. The motion compensated prediction 430 is shown straddling the boundary of the

sub-region 220. A portion 431 of the block residing outside of the sub-region 220 is not used for prediction and is padded by repeating the value of the pixel found at the edge of the sub-region 220. A portion 432 of the block residing in the sub-region 220 is used without any padding. A similar padding method is used for the rows and columns for blocks located at the vertical edge and horizontal edge, respectively.

[0039] Like the first embodiment, the method according to the second embodiment would also enable computational graceful degradation method to discard the portion of the picture outside the sub-region 220 without causing the encoder and decoder to drift apart.

[0040] Apart from motion compensation that may cause drift in inter blocks, intra blocks at the top and left boundary of the sub-region 220 are also restricted from using any blocks outside of the sub-region 220 for prediction. This is because in the computational graceful degraded decoder, these blocks would not be decoded and thus the prediction cannot be duplicated. This precludes the commonly used DC and AC coefficient prediction from being employed in the encoder.

[0041] Figure 2 also illustrates a block diagram of a decoder 190. The embodiment of the decoder 190 employing computational graceful degradation is described here. The compressed bitstream is received from the transmission and is passed to a Variable Length Decoder unit 191 where the bitstream is decoded according to the syntax and entropy method used. The decoded information is then passed to the Computational Graceful Degradation Selector 192 where the decoded information belonging to the sub-region 220 is retained and the decoded information outside of the sub-region 220 is discarded. The retained information is then passed to an Inverse Quantization unit 193 where the DCT coefficients are recovered. The recovered coefficients are then passed to an Inverse DCT unit 194 where the coefficients are transformed back to the spatial domain. The motion compensated prediction is then added to form the reconstructed picture. The reconstructed picture is stored in a Frame Storage 195 where it is used for the prediction of the next picture. A Motion compensation unit 196 performs the motion compensation according to the same method employed in the encoder 110.

[0042] In the first embodiment of the encoder where the motion vector is restricted no additional modification is required in the decoder. In the second embodiment of the encoder where the motion vector is not restricted, the motion compensation method with padding described above in connection with Fig. 5 is used in the decoder. Finally, intra blocks at the top and left boundary of the sub-region 220 are also restricted from using any blocks outside of the sub-region 200 for prediction. This precludes the commonly used DC and AC coefficient prediction from being employed.

[0043] In this embodiment the Complexity Parameter Encoder consist of a counting unit that counts the number of block decoding operations that are required. The block decoding operations are not basic arithmetic operations but rather a collection of operations that are performed on a block. A block decoding operation can be a block inverse quantization operation, a block inverse DCT operation, a block memory access or some other collection of operations that perform some decoding task on the block by block basis. The Complexity Parameter Encoder counts the number of blocks that require each set of operations and indicate these in the parameters. The reason block decoding operations are used instead of simple arithmetic operations is because different implementations may implement different operations more efficiently than others.

[0044] There is also a difference in decoder architecture and different amounts of hardware and software solutions that makes the use of raw processing power and memory access measures unreliable to indicate the complexity requirements. However, if the operations required are indicated by parameters that counts the number of block decoding operations necessary, the decoder can estimate the complexity. This is because the decoder knows the amount of operations required for each of the block decoding operations in its own implementation.

[0045] In the embodiment of the System Encoder and Multiplexer, the elementary bitstream are packetized and multiplexed for transmission. The information associated with the complexity parameters is also multiplexed into the bitstream. This information is inserted into the header of the packets. Decoders that do not require such information may simply skip over this information. Decoders that require such information can decode this information and interpret them to estimate the complexity requirements.

[0046] In this embodiment the encoder inserts the information in the form of a descriptor in the header of the packet. The descriptor contains an ID to indicate the type of descriptor it is followed by the total number of bytes contained in the descriptor. The rest of the descriptor contains the parameter for each of the block decoding operations. Optionally the descriptor may also carry some user defined parameters that are not defined earlier.

[0047] In the Scheduler and Complexity Estimator 180 in Figure 1, the time it takes for decoding all the audio-visual objects is computed based on the parameters found in the descriptor as well as the feedback information from the decoder.

[0048] An embodiment of the Complexity Estimator 180 is shown in Figure 6. The block decoding operation parameters 181a, 181b and 181c are passed into the complexity estimator 183 after being pre-multiplied with weightings 182a, 182b and 182c, respectively. The complexity estimator 183 then estimates the complexity of the picture to be decoder and passes the estimated complexity 184 to the decoder 190. After decoding the picture the decoder 190 returns the actual complexity

185 of the picture. An error 186 in the complexity estimation is obtained by taking a difference between the estimated complexity 184 and the actual complexity 185 of the picture. The error 186 is then passed into the feedback gain unit 187 where the corrections 188a, 188b and 188c to the weightings are found. The weights are then modified by these corrections and the process of estimating the complexity of the next picture continues.

[0049] The effect of this invention is that the need for implementations that can handle the worst case is no longer necessary. Using the indications of computational complexities and the computational graceful degradation methods simpler decoders can be implemented. The decoder would have the capabilities to decode most of the sequences, but if it encounters some more demanding sequences, it can degrade the quality and resolution of the decoder output in order to decode the bitstream.

[0050] This invention is also useful for inter working of services that have different resolutions and/or different formats. The sub-region can be decoder by the decoder of lower resolutions where as the decoder of higher resolutions can decode the full picture. One example is the inter working between 16:9 and 4:3 aspect ratio decoders.

Claims

1. A method for encoding a plurality of audiovisual objects into a compressed coded representation suitable for computational graceful degradation at the decoder comprising:

   encoding said audiovisual objects, incorporation methods allowing computational graceful degradation to be employed in the decoder, into their coded representations;
   estimating the implementation independent computational complexity measures in terms of a plurality of block decoding parameters;
   partitioning said coded representations of the audiovisual objects into a plurality of access units and adding header information to form packets;
   inserting a descriptor containing said block decoding parameters into the header of the packet; and
   multiplexing these packets to form a single multiplexed bitstream.

2. A method of decoding a multiplexed bitstream, with computational graceful degradation, to obtained a plurality of audiovisual objects, comprising:

   de-multiplexing the single multiplexed bitstream into a plurality of packets comprising of packet headers and access units;
   extracting the descriptor containing a plurality of block decoding parameters from the packet headers;
   reassembling the access units into their original coded representations of the audiovisual objects;
   estimating the decoder specific computational complexity measures based on said block decoding parameters and the current decoder implementation; and
   decoding said coded representations of the audiovisual objects, using computational graceful degradation, where necessary, to satisfy the estimated decoder specific computational complexity requirements.

3. A method of encoding said audiovisual objects into their compressed coded representations, according to claim 1, whereby incorporation methods allowing computational graceful degradation to be employed in the decoder, further comprises:

   partitioning the input pictures to be encoded into a plurality of sub-regions numbered in increasing order, beginning with the full picture as the first sub-region, where each sub-region comprising only of a subset of the pixels within the sub-region preceding it;
   entropy coding the position and dimension of the sub-regions into a compressed coded representation within the bitstream;
   further partitioning the sub-regions into a plurality of blocks for encoding into a compressed coded representation within the bitstream;
   performing motion estimation and motion compensation for said blocks using only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks;
   entropy coding the motion vectors into a compressed coded representation within the bitstream;
   transforming the motion compensated prediction difference into an orthogonal domain;
   quantizing the transformed coefficients using a quantization method; and
   entropy coding the quantized transformed coefficients into a compressed coded representation within the bitstream.

4. A method for decoding said coded representations of the audiovisual objects according to claim 2, using computational graceful degradation, where necessary, to satisfy the estimated decoder specific computational complexity requirements further comprising:

   entropy decoding the position and dimension of

the sub-regions from the compressed coded representation within the bitstream;
selecting only the blocks that are within the sub-region of interest for decoding;
entropy decoding the compressed coded representation to give quantized transformed coefficients;
inverse quantizing said quantized transformed coefficients to give the transformed coefficients;
inverse transforming said transform coefficients to give the spatial domain motion compensated prediction difference;
entropy decoding the motion vectors from the compressed coded representation within the bitstream;
performing motion compensation for said blocks using only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks; and
reconstructing the picture and storing said picture in the frame memory for prediction of the next picture.

5. A method of encoding said audiovisual objects into their coded representations, according to claim 1, whereby incorporation methods allowing computational graceful degradation to be employed in the decoder, further comprises:

partitioning the input pictures to be encoded into a plurality of sub-regions numbered in increasing order, beginning with the full picture as the first sub-region, where each sub-region comprising only of a subset of the pixels within the sub-region preceding it;
entropy coding the position and dimension of the sub-regions into a compressed coded representation within the bitstream;
further partitioning the sub-regions into a plurality of blocks for encoding into a compressed coded representation within the bitstream;
transforming said blocks into an orthogonal domain;
quantizing the transformed coefficients using a quantization method;
performing quantized transform coefficient prediction for said blocks using only the corresponding quantized transform coefficients from the blocks above and to the left that belong to sub-regions having the same or higher numeric order as said blocks; and
entropy coding the predicted difference of the quantized transformed coefficients into a compressed coded representation within the bitstream.

6. A method for decoding said coded representations of the audiovisual objects according to claim 2, using computational graceful degradation, where necessary, to satisfy the estimated decoder specific computational complexity requirements further comprising:

entropy decoding the position and dimension of the sub-regions from the compressed coded representation within the bitstream;
selecting only the blocks that are within the sub-region of interest for decoding;
entropy decoding the compressed coded representation to give quantized transformed coefficients;
performing quantized transform coefficient prediction for said blocks using only the corresponding quantized transform coefficients from the blocks above and to the left that belong to sub-regions having the same or higher numeric order as said blocks;
inverse quantizing said quantized transformed coefficients to give the transformed coefficients;
inverse transforming said transform coefficients to give the spatial domain pixel values; and
reconstructing the picture and storing said picture in the frame memory for prediction of the next picture.

7. A method for estimating the implementation independent computational complexity measures, as in claim 1, whereby the plurality of block decoding parameters comprises numeric numbers indicating the number of

block entropy decoding operations;
block motion compensation operation;
block inverse quantization operations;
block transform operations;
block addition operations; and
block memory access operations.

8. A method of encoding the block decoding parameters in the header of the packet as in claim 1, where the descriptor comprises:

a descriptor identification number signaling the descriptor type;
a descriptor length field to indicate the size of the descriptor; and
a plurality of block decoding parameters.

9. A method of partitioning the input pictures to be encoded into a plurality of sub-regions according to claims 3 and 5, where said sub-regions are rectangular.

10. A method of performing motion estimation and motion compensation for said blocks according to claim 3, whereby using only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks, implies that only prediction blocks that lie completely within said sub-regions are selected.

11. A method of performing motion estimation and motion compensation for said blocks according to claim 3, whereby using only the pixels from the reconstructed picture that belong to sub-regions having the same or higher numeric order as said blocks, implies that prediction blocks may lie partially outside said sub-regions but with the additional condition that the pixels lying outside said sub-region are replaced by the nearest pixels from within the sub-regions.

12. A method of partitioning the pictures into a plurality of sub-regions according to claims 3 and 5, where the position and dimension of each of said sub-regions may vary from picture to picture and said position and said dimension are coded by means of a pan scan vector, giving the horizontal and vertical displacement, a width and a height.

13. A method of partitioning the pictures into a plurality of sub-regions according to claims 3 and 5, where the position and dimension of the sub regions are the same from picture to picture and said position and said dimension are coded once at the beginning of the sequence by means of a horizontal and vertical displacement, a width and a height.

14. A method of encoding and decoding according to claims 3, 4, 5 and 6, where the transform is the Discrete Cosine Transform.

15. A method of encoding and decoding according to claims 3, 4, 5 and 6, where the number of sub-regions is two.

16. A method where there is plurality of sub-region numbered in increasing order and the motion vector can point into a sub region of lower order but not out of a lower order to a higher ordered number.

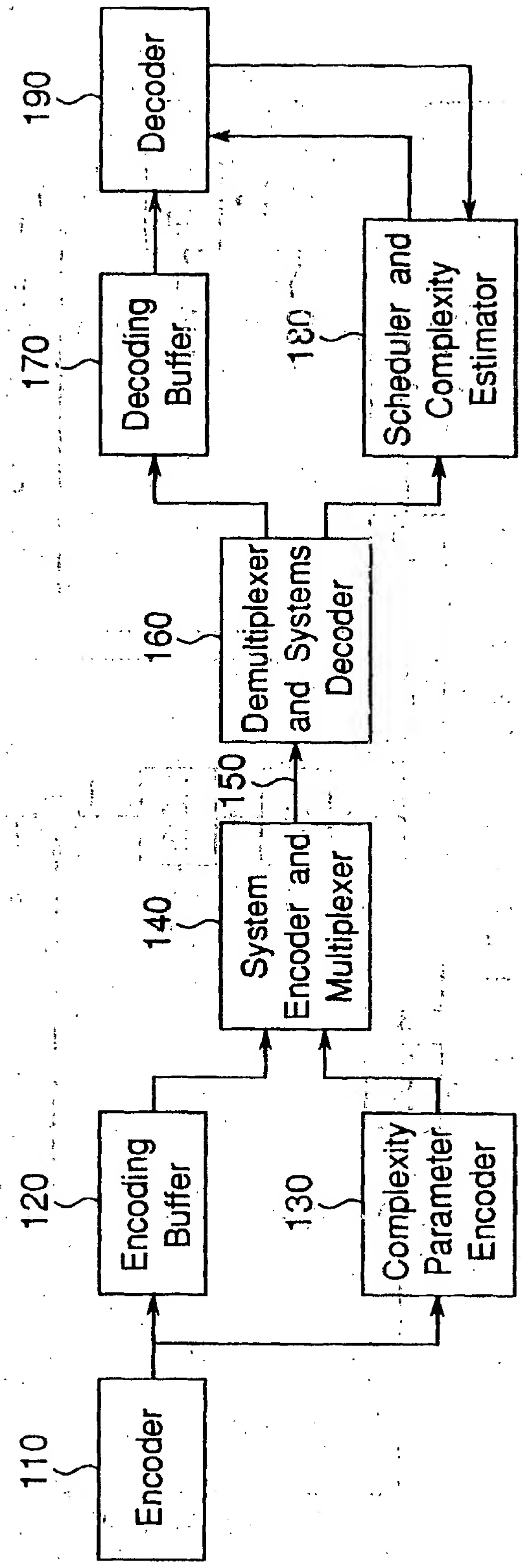
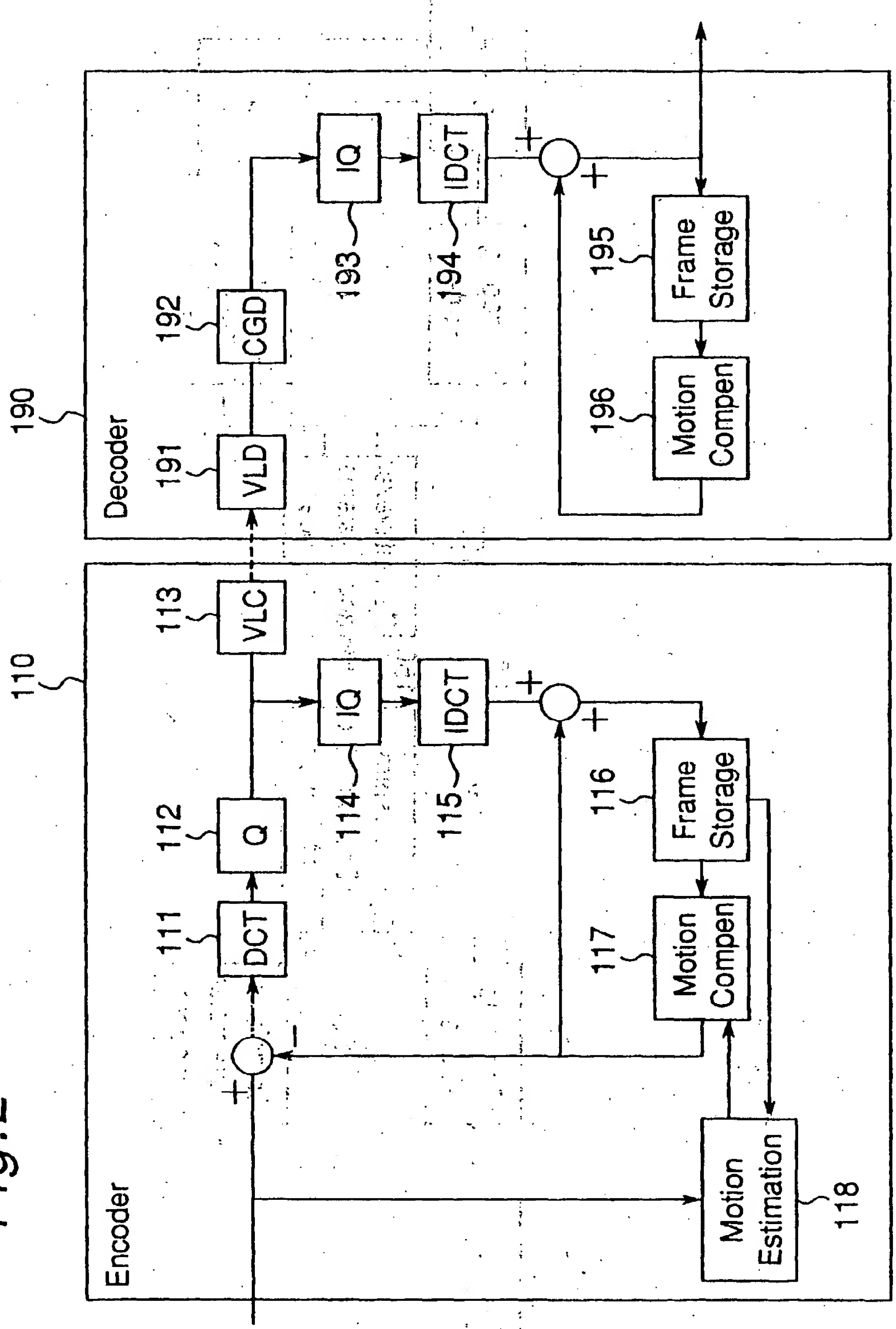

110
Encoder
120
Encoding Buffer
130
Complexity Parameter Encoder
140
System Encoder and Multiplexer
150
160
Demultiplexer and Systems Decoder
170
Decoding Buffer
180
Scheduler and Complexity Estimator
190
Decoder

Fig. 1

Fig.2



Encoder
110
111
DCT
112
Q
113
VLC
IQ
114
IDCT
115
Frame Storage
116
Motion Compen
117
Motion Estimation
118
Decoder
190
191
VLD
192
CGD
IQ
193
IDCT
194
Frame Storage
195
Motion Compen
196

Fig.3

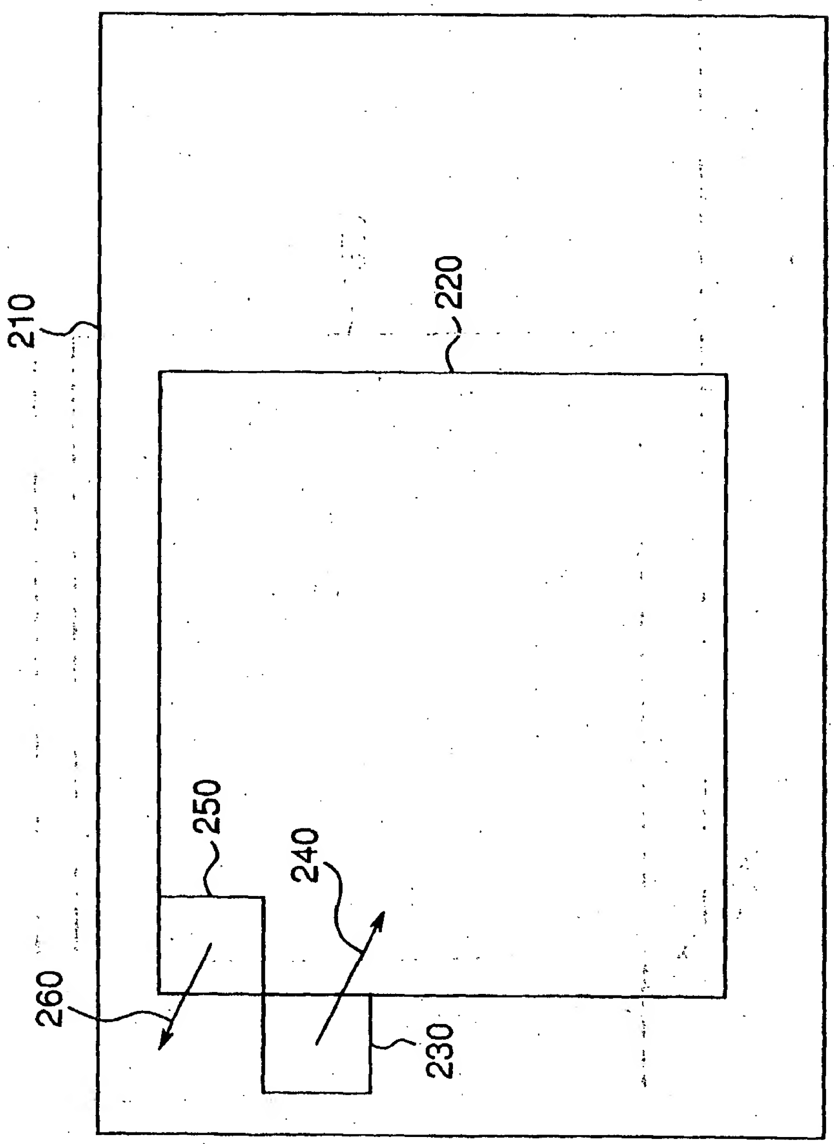


Fig.4

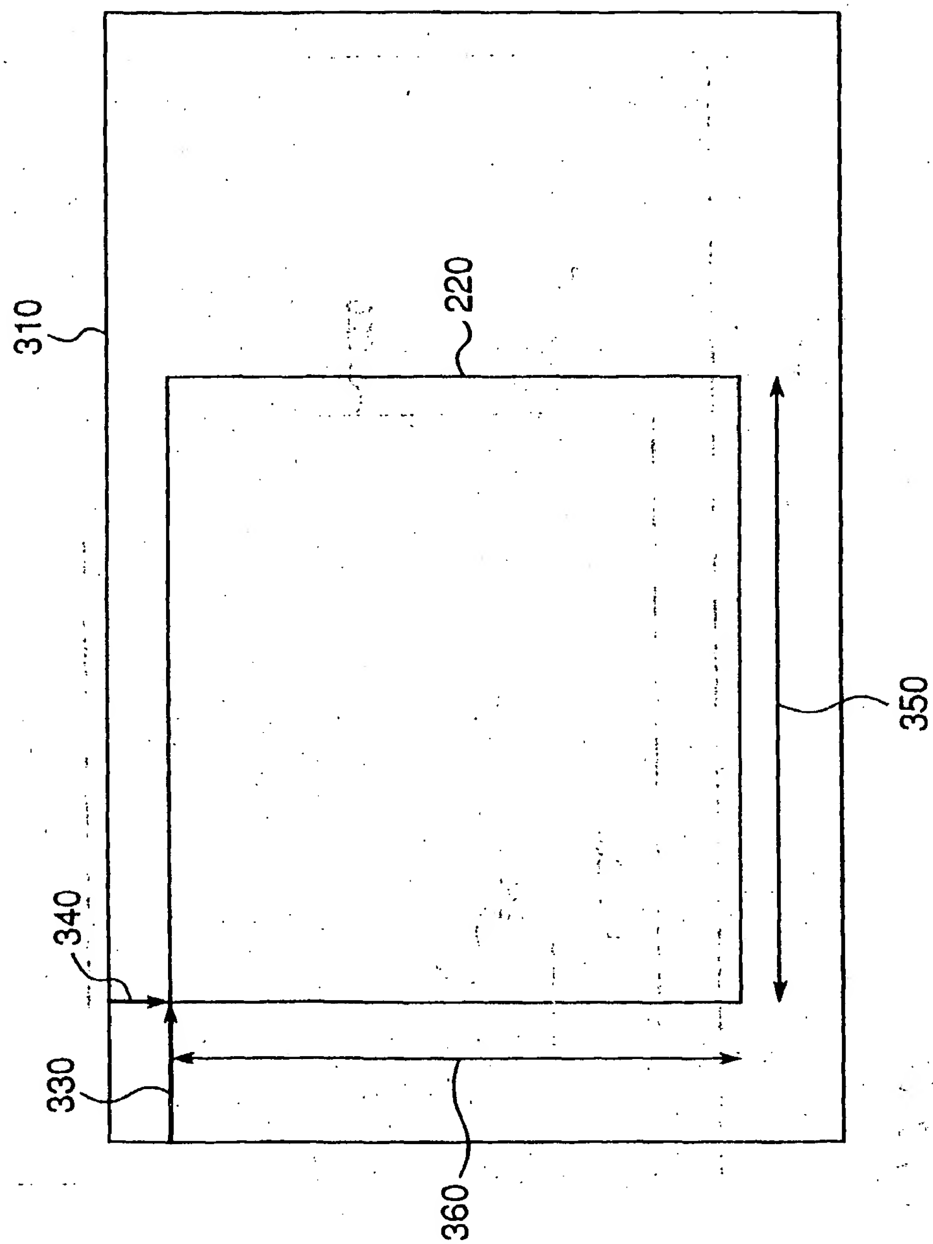


Fig.5

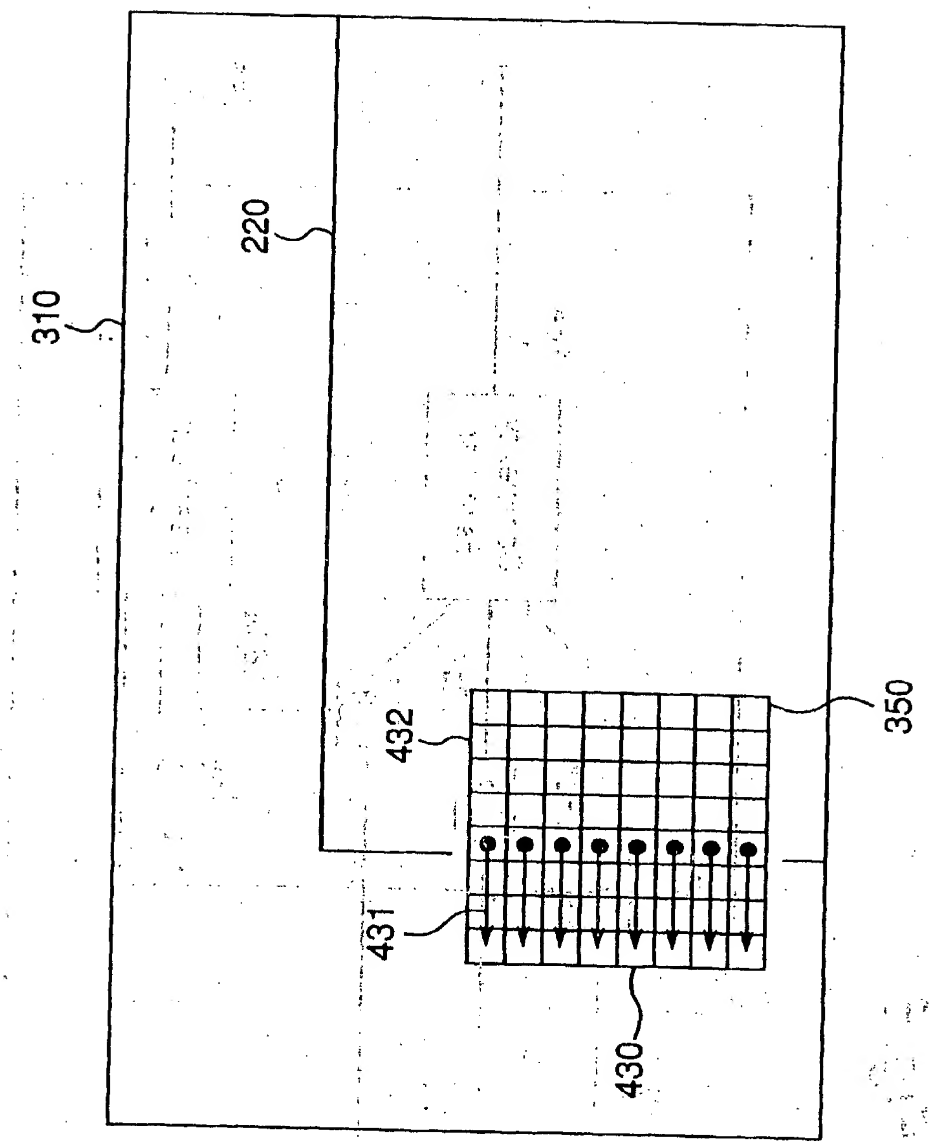
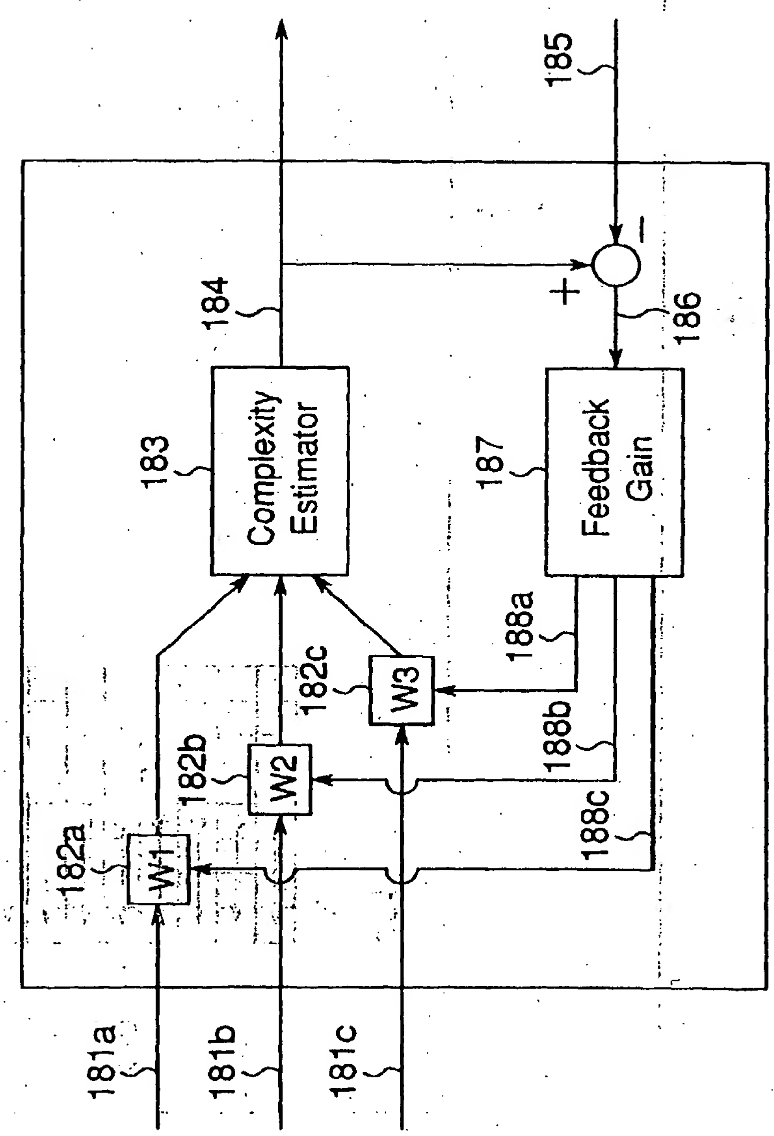


Fig.6


Complexity Estimator
Feedback Gain
W1
W2
W3
181a
181b
181c
182a
182b
182c
183
184
185
186
187
188a
188b
188c
+
−

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 912 063 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
08.03.2000 Bulletin 2000/10

(43) Date of publication A2:
28.04.1999 Bulletin 1999/17

(21) Application number: 98119908.6

(22) Date of filing: 21.10.1998

(51) Int. Cl.[7]: **H04N 7/24**, H04N 7/26, H04N 7/36, H04N 7/50

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 24.10.1997 SG 9703861

(71) Applicant:
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.
Kadoma-shi, Osaka 571-8501 (JP)

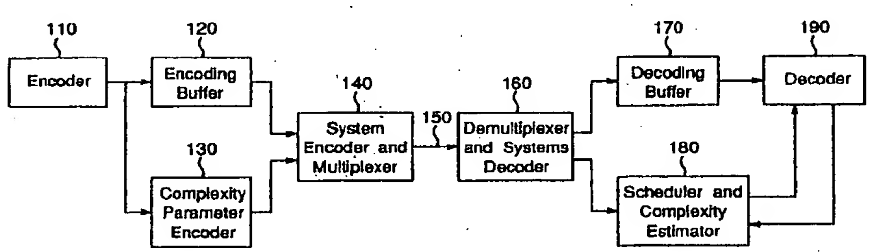
(72) Inventor: Tan, Thiow Keng
08-506 Singapore 470601 (SG)

(74) Representative:
Eisenführ, Speiser & Partner
Martinistrasse 24
28195 Bremen (DE)

(54) A method for computational graceful degradation in an audiovisual compression system

(57) The invention disclosed here is a method for an encoder to encode audiovisual information for transmission to the decoder without any prior knowledge of the computational capabilities of the decoder. A descriptor containing parameters that can be used to estimate the complexity of the decoding process is embedded in the system stream. The encoder also encodes the video information in such a manner that the decoder can choose to ignore some of the information and only decode a subset of the encoded information in order to reduce the computational requirements. This method allows more than one decoder to decode the same bitstream giving different resolutions depending on the computational capability of the decoder.

Fig. 1



European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 11 9908

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X A	EP 0 676 899 A (AT & T CORP) 11 October 1995 (1995-10-11) * column 1, line 55 - column 2, line 39 *	1,2 3-15	H04N7/24 H04N7/26 H04N7/36 H04N7/50
X Y A	EP 0 650 298 A (SONY CORP) 26 April 1995 (1995-04-26) * page 4, line 26 - page 10, line 14 * * page 13, line 42 - page 13, line 44 * * figure 4 *	16 1-6, 8-10, 12-15 7,11	
Y	US 5 402 146 A (PIETRAS MARK A ET AL) 28 March 1995 (1995-03-28) * column 3, line 64 - column 5, line 17 * * figure 1 *	1-6, 8-10, 12-15	
A	EP 0 677 961 A (TOSHIBA AVE KK ;TOKYO SHIBAURA ELECTRIC CO (JP)) 18 October 1995 (1995-10-18) * column 2, line 42 - column 3, line 49 * * figures 1-3,13-16 *	1-15	TECHNICAL FIELDS SEARCHED (Int.Cl.6) H04N
A	EP 0 739 141 A (TOKYO SHIBAURA ELECTRIC CO) 23 October 1996 (1996-10-23) * page 4, line 26 - page 4, line 49 * * figures 1,5A,5B * * claims 1-3 *	1-15	
E	WO 99 12126 A (AHN CHIE TEUK ;PARK SANG KYU (KR); LIM YOUNG KWON (KR); KOREA ELEC) 11 March 1999 (1999-03-11) * the whole document *	1,2	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	14 January 2000	Fassnacht, C

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)